Underwater acoustic signal classification based on sparse time-frequency representation and deep learning

Yongchun Miao, Yuriy V. Zakharov, *Member, IEEE,* Haixin Sun, *Member, IEEE,* Jianghui Li, *Member, IEEE* and Junfeng Wang

Abstract

For classification of underwater acoustic signals, we propose a novel sparse anisotropic chirplet transform (ACT) to reveal fine time-frequency structures. The signal features in the form of a time-frequency map are fed into a deep convolutional neural network, referred to as a time-frequency feature network (TFFNet), which brings flexibility to signal classification. TFFNet is based on a novel efficient feature pyramid enhancing feature maps by aggregating the context information at different scales. To remove the gridding artefacts on enhanced feature maps, a form of aggregating transformation, a forward feature fusion, is utilized to merge the forward feature maps. Main contributions of this work are a novel sparse ACT, an TFFNet classifier and an efficient feature pyramid (EFP) with forward feature fusion. Experimental results demonstrate that the sparse ACT provides a high-resolution time-frequency representation of underwater signals and the TFFNet improves the classification performance compared to known networks and two machine learning methods (random forest and support vector machine with radial basis function kernel) on two real datasets, an underwater acoustic communication signal dataset and whale sounds dataset.

Index Terms

Deep convolutional neural network (CNN), efficient feature pyramid (EFP), sparse anisotropic chirplet transform (ACT), time-frequency representation (TFR), underwater acoustic signal classification.

I. Introduction

UNDERWATER environment is characterized by many acoustic signals, such as biological signals, communication signals, seismic signals, etc. Acoustic event detection requires automatic identification to distinguish classes of the signals. In marine applications, signals are usually observed in noisy environments [1]. These environments pose the challenge for the signal classification.

To extract features from sound data for classification, efficient methodes have been presented, such as methods based on the mel frequency cepstral coefficients (MFCCs) [2], matched filtering [3], [4], classical energy spectrum analysis [5] and others. However, when the signal is noisy or overlapped with other signals, the features become smeared in the time/frequency domain. Recently, alternative features have been used, which are provided by time-frequency (TF) distributions such as the short time Fourier transform (STFT), wavelet transform (WT), chirplet transform (CT) [6], [7], etc. However, they are sensitive to noise or suffer from inability to represent overlapped components. Inspired by the approach of the TF reassignment [8], Fourier synchrosqueezed transform (FSST) [9], sparse CT [10], [11] and anisotropic CT (ACT) [12] have been proposed to obtain highly concentrated time-frequency representations (TFRs). Since the reassignment vector for an underwater acoustic (UWA) signal with high sampling rate can be large, the complexity of such techniques is often unacceptable. Thus, it is essential to develop less complicated transforms of UWA signals into TF feature vectors, which could also improve the signal classification.

Previous works considering the classification of UWA signals are primarily based on traditional "shallow" machine learning architectures [13], the support vector machines [14], [15] and gradient boosting decision tree [16]. The drawback of these methods is their low capacity so that the addition of more training data cannot always improve the classification performance [17]. However, the deep learning can help in this situation [18], [19]. The deep convolutional neural networks (CNNs) are becoming popular in image processing and computer vision applications, and they are now increasingly and successfully used on acoustic signal sets [20]–[24].

Two general approaches are most often used to classify signals based on deep CNNs: 1) utilizing TF transforms for adopting the transfer learning on a pretrained CNN, similar to the process of image classification [20]; 2) extracting a set of specific signal features as direct inputs to CNN classifiers for training model [21]–[23]. Both these approaches are capable of automatically learning multiscale feature maps and delivering better performance results when amount of data increases. Some networks, e.g. fully convolutional networks (FCN) [25], Region-based FCN (R-FCN) [19], Fast R-CNN [26] and Faster R-CNN [27]

Yongchun Miao and Haixin Sun are with the School of Informatics, Xiamen University, Xiamen, Fujian, 361005, China (e-mail: ycmiao@stu.xmu.edu.cn and hxsun@xmu.edu.cn)

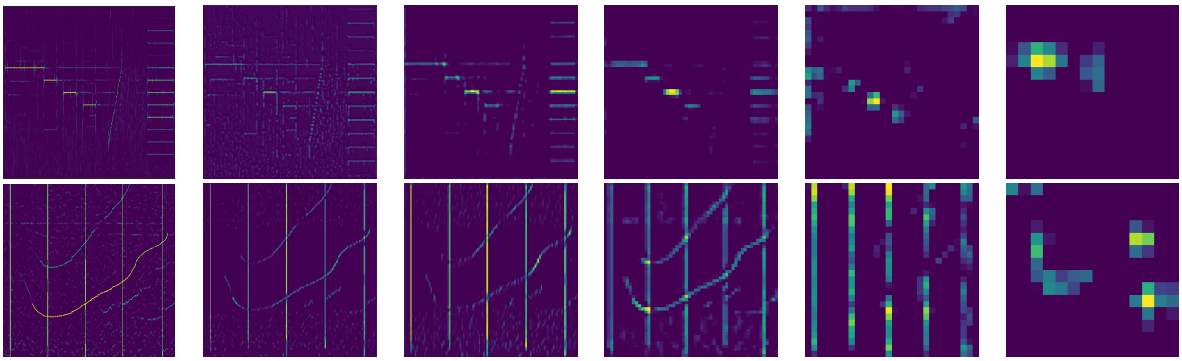
Yuriy V. Zakharov is with the Department of Electronics, University of York, York YO10 5DD, U.K. (e-mail:yury.zakharov@york.ac.uk)

Jianghui Li is with the Institute of Sound and Vibration Research, University of Southampton, Southampton, SO17 1BJ, U.K. (e-mail: J.Li@soton.ac.uk)

Junfeng Wang is with the School of Electrical and Electronic Engineering, Tianjin University of Technology, Tianjin, 300384, China (e-mail: great_seal@163.com)

Corresponding author: Haixin Sun and Junfeng Wang

exploit multiple feature hierarchies to improve the classification performance. While the accuracy of these networks can be improved with network's size in depth, the network complexity also increases. The single-shot multi-box detector (SSD) [28] that uses the visual geometry group network (VGG-16) as a feature pyramid (FP) has demonstrated its success in reducing the computational complexity, while providing the classification performance comparable to that of the region-based networks. However, the classifier with this FP would poorly perform on the deeper layers because of the lack of component-level information.



(a) Input (b) $C_{1,2}$ (c) $C_{2,2}$ (d) $C_{3,3}$ (e) $C_{4,3}$ (f) $C_{5,3}$

Fig. 1. Feature maps of two signals when using the SSD with a VGG-16 [18] backbone. (a) TFRs of an underwater communication signal (top TFRs: $t \in [0, 0.74]$ s, $f \in [6, 20]$ kHz) and a whale signal (bottom TFRs: $t \in [0, 1.0417]$ s, $f \in [0, 20]$ kHz) (b)∼(f) Outputs of convolution layers.

This problem is especially prominent when inputs to a network are TF images. For example, Fig. 1 shows TFRs of two UWA signals and their feature maps when using the SSD with a VGG-16 [18] network. The shallower layers (closer to the input, i.e., $C_{1,2}$ and $C_{2,2}$) learn very generic features like edges, corners and so on, while deeper layers (closer to the output layer, $C_{4,3}$ and $C_{5,3}$) learn abstract features. Although the simplified representation of the deeper layers can ensure pixel invariants independent of the illumination, noise, etc., it can cause difficulties for thin or linear structures in the TF image. Since a FP is able to preserve rich contextual information of these structures at all scales, the network aggregates the contextual information to generate feature maps of TF images.

Deep learning networks are capable of learning valuable differences between similar classes of UWA signals automatically, which may promote our understanding of the UWA in the future. Therefore, it is essential to make additional effort for better representation of UWA signals and develop superior networks.

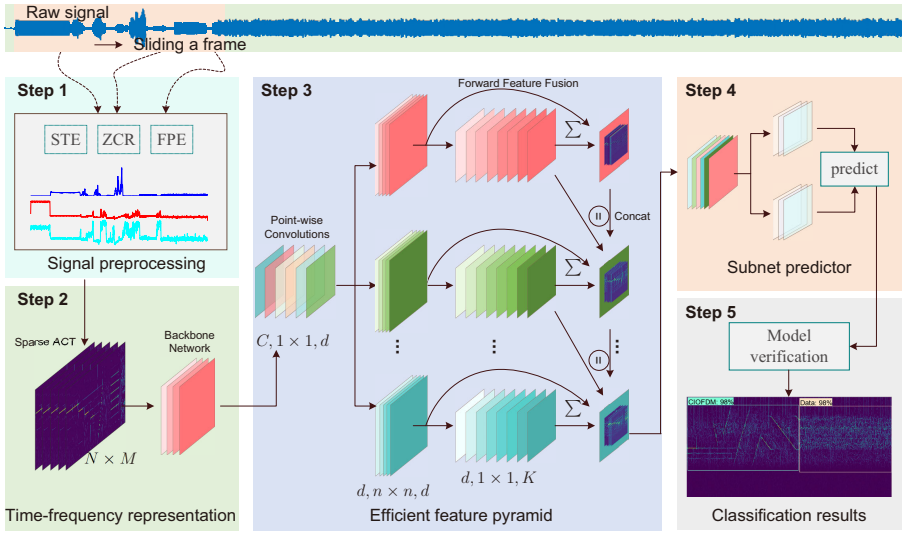


Fig. 2. The framework of the UWA classification based on deep learning and TF feature

The main contributions of this work are as follows (see Fig. 2).

1) A sparse anisotropic chirplet transform (ACT) to represent the fine TF structure of acoustic signals is proposed. It provides features for classification of sounds that are commonly considered difficult to classify using known transforms.
2) A deep convolutional neural network architecture, called the TF feature network (TFFNet), which is well suited for classification of UWA signals, is designed.

3) An efficient feature pyramid (EFP) with forward feature fusion is proposed for reducing the network complexity and concatenating the context information at different scales.
4) The proposed TFFNet classifier is tested on two UWA data sets, underwater acoustic communication signals and whale signals to demonstrate its superior performance against state-of-the-art network classifiers.

The rest of this paper is organized as follows. Section II introduces the CT and sparse ACT. The TFFNet architecture based on EFP is elaborated in Section III. The analysis of real data is discussed in Section IV. Finally, conclusions are drawn in Section V.

II. Sparse ACT

A. Sparse ACT and its derivatives

Here, we present the CT combined with the reassignment method for obtaining a sparse TFR. The CT [6] of an analytical signal $x(t)$ is defined as

$$S(t,f)=\int_{-\infty}^{\infty}x(t+\tau)h(\tau)e^{-j2\pi\frac{c}{2}\tau^2}e^{-j2\pi f\tau}\mathrm{d}\tau, \tag{1}$$

where c is the chirp rate. The smoothing window in (1) is the unit-energy Gaussian function $h(t)=\sigma^{-1/2}\pi^{-1/4}e^{-t^2/2\sigma^2}$, where σ defines the window width. A drawback however is that for a multicomponent signal, the window $h(t)$ does not allow a high TF resolution of all components of the signal.

Considering different standard deviations (e.g. σ_t and σ_f) in time and frequency, the time and frequency resolution of the TFR can be controlled simultaneously. An anisotropic Gaussian window [12] provides more flexibility for the TFR of multicomponent signals. The anisotropic Gaussian window is given by

$$\begin{aligned}\hat{h}(t,f)=\frac{1}{\sqrt{\pi\sigma_t\sigma_f}}\exp\Bigg\{-\frac{1}{2}\Bigg(&\frac{(\lambda(t\cos\theta+f\sin\theta))^2}{\sigma_t^2}\\&+\frac{(\lambda^{-1}(-t\sin\theta+f\cos\theta))^2}{\sigma_f^2}\Bigg)\Bigg\},\end{aligned} \tag{2}$$

where $\lambda\geq 0$ is a parameter controlling the degree of anisotropy, by introducing different scaling along the t- and f-axis, and θ is an angle determined from [29]

$$\Im\left\{\eta(t,f)\left(\frac{\lambda^2}{\sigma_t^2}\cos\theta+j\sin\theta\right)\right\}=0, \tag{3}$$

where $\eta(t,f)=\hat{S}(t,f)/S(t,f)$, and the transform

$$\hat{S}(t,f)=\frac{1}{\sigma_t^2}\int_{-\infty}^{\infty}\tau x(t+\tau)h(\tau)e^{-j2\pi\frac{c}{2}\tau^2}e^{-j2\pi f\tau}\mathrm{d}\tau \tag{4}$$

is based on (1), but with $h(t)$ replaced by $\mathrm{d}h(t)/\mathrm{d}t$. For brevity, (2) can be written as

$$\hat{h}(t,f)=h_{\sigma_t}(\tilde{t})h_{\sigma_f}(\tilde{f}), \tag{5}$$

where $\tilde{t}=\lambda(t\cos\theta+f\sin\theta)$ and $\tilde{f}=\lambda^{-1}(-t\sin\theta+f\cos\theta)$.

With the window $\hat{h}(t,f)$ instead of the window $h(t)$ (1), an anisotropic CT (ACT) $\tilde{S}_{\sigma_t\sigma_f}(t,f)$ can be obtained as

$$\begin{aligned}\tilde{S}_{\sigma_t\sigma_f}(t,f)&=\iint_{\mathbb{R}^2}S(\tau,v)\hat{h}(\tau-t,v-f)\mathrm{d}\tau\mathrm{d}v,\\&=\mathcal{M}(\tilde{t},\tilde{f})e^{j\phi(\tilde{t},\tilde{f})},\end{aligned} \tag{6}$$

where $\mathcal{M}(\tilde{t},\tilde{f})=\left|\tilde{S}_{\sigma_t\sigma_f}(t,f)\right|$ and $\phi(\tilde{t},\tilde{f})$ are the magnitude and phase of the CT, respectively, and $S(t,f)$ is the CT given by (1).

To improve the readability of TFR, a TF reassignment vector is introduced on the spectrogram of the ACT to sharpen the TF representation [30], which reassigns a point $(\tilde{t},\tilde{f})$ to a point of maximum energy contribution $(\hat{t}(\tilde{t},\tilde{f}),\hat{f}(\tilde{t},\tilde{f}))$. The reassignment on the spectrogram (6) is used to compensate for the time and frequency shifts induced by the smoothing window in (5).

The TF reassignment $(\tilde{t},\tilde{f})\rightarrow(\hat{t}(\tilde{t},\tilde{f}),\hat{f}(\tilde{t},\tilde{f}))$ depends on the gradient of the CT phase [30], [31]:

$$\begin{aligned}\hat{t}(\tilde{t},\tilde{f})&=\tilde{t}/2-\partial_{\tilde{f}}\phi(\tilde{t},\tilde{f}),\\\hat{f}(\tilde{t},\tilde{f})&=\tilde{f}/2+\partial_{\tilde{t}}\phi(\tilde{t},\tilde{f}).\end{aligned} \tag{7}$$

The phase gradient can be obtained as [30]

$$\begin{aligned}\partial_{\tilde{t}}\phi(\tilde{t},\tilde{f}) &= -\Im\left(\eta(\tilde{t},\tilde{f})\right)+\tilde{f}/2,\\ \partial_{\tilde{f}}\phi(\tilde{t},\tilde{f}) &= -\sigma_t^2\Re\left(\eta(\tilde{t},\tilde{f})\right)-\tilde{t}/2,\end{aligned} \tag{8}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote real and imaginary parts of a complex number. By substituting (8) in (7), the reassignment can be achieved by the time and frequency shifts:

$$\begin{aligned}\hat{t}(\tilde{t},\tilde{f})-\tilde{t} &= \sigma_t^2\Re\left(\eta(\tilde{t},\tilde{f})\right),\\ \hat{f}(\tilde{t},\tilde{f})-\tilde{f} &= -\Im\left(\eta(\tilde{t},\tilde{f})\right).\end{aligned} \tag{9}$$

Based on the TF reassignment, the generalized sparse anisotropic chirplet transform (ACT) can be defined as

$$\hat{T}(\tilde{t},\tilde{f}) = \iint_{\mathbb{R}^2}\tilde{S}_{\sigma_t}(\tau,v)\kappa_\theta(\tau-\tilde{t},v-\tilde{f})\mathrm{d}\tau\mathrm{d}v, \tag{10}$$

where $\tilde{S}_{\sigma_t}(\tilde{t},\tilde{f})$ is given by (1) with a window $h_{\sigma_t}(\tilde{t})$ and $\kappa_\theta(\tilde{t},\tilde{f}) = \frac{1}{\cos^2\theta}\delta(\tilde{t}-\hat{t}(\tilde{t},\tilde{f}))\delta(\tilde{f}-\hat{f}(\tilde{t},\tilde{f}))$ is a 2D kernel.

The sparse ACT allows modelling the energy concentration that is direction- and position-dependent (anisotropic) and is particularly well adapted to multicomponent signals. However, the reassignment process for the sparse ACT has a higher computational complexity than that for the ACT.

B. Fast implementation of sparse ACT

To simplify the reassignment process, we change the coordinate system as in [32] such that the fast implementation procedure for sparse ACT introduced in Algorithm 1 can be used. Since the analytical signal $x(t)$ with a sampling frequency F_s consists of multiple components, Algorithm 1 obtains TFRs for N_σ different standard deviations σ_t, σ_f and N_θ different angles θ_{scale}. **CT**$(\cdot)$ denotes a function that obtains a TFR of the chirplet transform using (1). Depending on the value of $\eta(t,f) = \hat{S}(t,f)/S(t,f)$, contour preference angles v for TFRs are determined in (3). A function **Findridges**$(\cdot)$ with parameters v, orientations $\tan\vartheta$ in (13) and directional filters [-1,0,1] [33] is called for directional ridge detection in the TF plane. N_q ridges are found at TF points by local approximation [33] with two-dimensional gradient (8) of TFR. For each ridge $q_{n_q}, n_q = 1,\cdots,N_q$, the reassignment process of sparse ACT is then performed to obtain TFR $\tilde{T}$. Since a TFR matrix $\tilde{T}$ of sparse ACT contains many zeros, a **Sparse**$(\cdot)$ function converts the TFR matrix to sparse storage to save memory. Finally, an enhanced TFR from Algorithm 1 combines TFRs across variations in σ_t, σ_f and angle. The fast reassignment process of sparse ACT is implemented as follows.

The tf-axes coordinate system is transformed into a non-orthogonal $t\omega$-axes system [32],

$$\hat{h}(t,f) = \tilde{h}(t,\omega), \tag{11}$$

where $\omega = \sqrt{t^2+f^2}$. In the $t\omega$-axes system, since the Fourier transform of $\tilde{h}(t,\omega)$ can be written as a product of two functions, $H(\omega_t,\omega_\omega) = H_{\sigma_\tau}(\omega_t)H_{\sigma_\omega}(\omega_\omega)$, where $H_{\sigma_\tau}(\omega_t)$ and $H_{\sigma_\omega}(\omega_\omega)$ express the Fourier transform of $h_{\sigma_\tau}(t)$ and $h_{\sigma_\omega}(\omega)$, respectively, the anisotropic Gaussian window $\tilde{h}(t,\omega)$ can be decomposed into two subsequent convolutions $\tilde{h}(t,\omega) = h_{\sigma_\tau}(t) * h_{\sigma_\omega}(\omega)$.

The first factor represents a one-dimensional Gaussian function in the t-direction with a standard deviation σ_τ,

$$h_{\sigma_\tau}(t) = \frac{1}{\sqrt{\sqrt{\pi}\sigma_\tau}}e^{-\frac{t^2}{2\sigma_\tau^2}}. \tag{12}$$

The second one-dimensional Gaussian function along the line $f = t\tan\vartheta$, where

$$\tan\vartheta = \frac{(\lambda^{-1}\sigma_f)^2\cos^2\theta+(\lambda\sigma_t)^2\sin^2\theta}{((\lambda\sigma_t)^2-(\lambda^{-1}\sigma_f)^2)\cos\theta\sin\theta}, \tag{13}$$

is expressed as

$$h_{\sigma_\omega}(\omega) = \frac{1}{\sqrt{\sqrt{\pi}\sigma_\omega}}e^{-\frac{\omega^2}{2\sigma_\omega^2}}. \tag{14}$$

The time-scale σ_τ and frequency-scale σ_ω are respectively given by

$$\sigma_\tau = \frac{\sigma_t\sigma_f}{\sqrt{(\lambda^{-1}\sigma_f)^2\cos^2\theta+(\lambda\sigma_t)^2\sin^2\theta}},$$

$$\sigma_\omega = \sqrt{(\lambda^{-1}\sigma_f)^2\cos^2\theta+(\lambda\sigma_t)^2\sin^2\theta}.$$

Algorithm 1: Fast sparse ACT algorithm

Input: x, F_s, $\sigma_t = \sigma_f = 0.2 : 0.3 : 2.6$,
$\theta_{scale} = (\pi/8 : \pi/8 : \pi) + \pi/8$, $\lambda = 4$
Output: $\hat{T}$

for $n_\sigma = 1 : N_\sigma$ **do**
  $\sigma_v^2 = \lambda \sigma_t^2(n_\sigma)$
  $\sigma_u^2 = \sigma_f^2(n_\sigma)/\lambda$
  for $n_\theta = 1 : N_\theta$ **do**
    $\theta = \theta_{scale}(n_\theta)$
    $\alpha = \sigma_v^2 \cos^2(\theta) + \sigma_u^2 \sin^2(\theta)$
    $\beta = (\sigma_v^2 - \sigma_u^2)\cos(\theta)\sin(\theta)$
    $\gamma = \sigma_v^2 \sin^2(\theta) + \sigma_u^2 \cos^2(\theta)$
    $\sigma_\tau = \sqrt{(\alpha - \beta^2/\gamma)}$
    $\tan\vartheta = \frac{\beta}{\gamma}$
    $h = \frac{1}{\sqrt{\sqrt{\pi}\sigma_\tau}} e^{-\frac{t_w^2}{2\sigma_\tau^2}}$
    $\mathrm{d}h = -2h\frac{t_w}{\sigma_\tau^2}$
    $S =$**CT**(x, h, F_s)
    $\hat{S} =$**CT**$(x, \mathrm{d}h, F_s)$
    $\eta = \frac{\hat{S}}{S}$
    $v = \Im\left\{\eta\left(\frac{\lambda^2}{\sigma_t^2(n_\sigma)}\cos\theta + j\sin\theta\right)\right\}$
    $q =$**Findridges**$(v, \tan\vartheta, [-1, 0, 1])$
    for $n_q = 1 : N_q$ **do**
      Find indices i_{index} of pixels on n_q-th directional ridge q_{n_q}
      $\tilde{T}(i_{\mathrm{index}}) = \sum(|(S(i_{\mathrm{index}})|)$
    end
    $T(n_\sigma, n_\theta)$=**Sparse**$(\tilde{T})$
  end
end
for $n_\sigma = 1 : N_\sigma$ **do**
  for $n_\theta = 1 : N_\theta$ **do**
    if $n_\theta = 1$ **then**
      T=$T(n_\sigma, 1)$ +$T(n_\sigma + 1, n_\theta)$ + $T(n_\sigma, N_\theta)$
    else
      T=$T(n_\sigma, n_\theta)$ +$T(n_\sigma + 1, n_\theta)$ + $T(n_\sigma, n_\theta - 1)$
    end
    $\hat{T} = \hat{T} + T$
  end
end
return $\hat{T}$

Substituting (12) and (14) into (11), the kernel is represented as:

$$\hat{h}(t, f) = h_{\sigma_\tau}(t) * h_{\sigma_\omega}(\sqrt{t^2 + f^2}). \tag{15}$$

In the $t\omega$-axes system, according to (15), the implementation of the sparse ACT in (10) comes down to first applying the CT with a window function $h_{\sigma_\tau}(t)$. The resulting TFR $S_{\sigma_\tau}(t, f)$ is then smoothed with a Gaussian window $h_{\sigma_\omega}(\omega)$ along the line $f = t\tan\vartheta$. For fast implementation of the reassignment process, the value of the source pixel is obtained by interpolating between pixels at the line of interest [32]. Thus, the fast implementation of sparse ACT (10) is approximated by

$$\begin{aligned}\tilde{T}(n, m) = \sum_{k=1}^{M} \{ & S_{\sigma_\tau}(\lfloor n - k\tan\vartheta \rfloor, m - k) \\ & + S_{\sigma_\tau}(\lfloor n + k\tan\vartheta \rfloor, m + k) \\ & + S_{\sigma_\tau}(\lfloor n - k\tan\vartheta - 1 \rfloor, m - k) \\ & + S_{\sigma_\tau}(\lfloor n + k\tan\vartheta + 1 \rfloor, m + k)\},\end{aligned} \tag{16}$$

where M is the number of sampled points on the line $f = t\tan\vartheta$, the symbol $\lfloor\cdot\rfloor$ indicates rounding down, and $S_{\sigma_\tau}(t, f)$ is obtained by the CT (1). Notice that the $f \pm k$ coordinate falls exactly on the line $f = t\tan\vartheta$, whereas the $t \pm k\tan\vartheta$ may fall between two pixels. The linear interpolation is used to compute samples between the neighbouring pixels at $t \pm k\tan\vartheta$.

The TF image $T(n, m) \in \mathbb{R}^{N\times M}$ with the time span N and frequency span M is used as an input to the learning network.

III. TFFNET

To enhance TF feature maps, we introduce an efficient feature pyramid (EFP) module for leveraging the convolution factorization to design a TF feature network (TFFNet) architecture. Table. I summarizes the TFFNet architecture. In the EFP, convolutional layers are denoted as "d (layer number)_conv (kernel size)_(dilation rate)". A basic EFP module has 5 parallel layers that apply 3×3 convolutions with dilation rates 1, 2, 4, 8, 16, respectively. The number of feature maps of four branch layers are $d_1 = d_2 = d_3 = d_4 = d = \lfloor \frac{C}{5} \rfloor$ and the number of feature maps of the last branch layer is $d_5 = C - 4d$. The EFP and forward feature fusion considered below in Sections III-A and III-B, are designed to improve the performance of the TFFNet by aggregating multi-scale contextual information.

To predict object classes and bounding boxes of the predicted class, TFFNet adopts a subnetwork predictor following the opinion of Faster R-CNN [26] that applies bounding-box classification and regression in parallel. Before each feature map from the EFP is fed into a sequence of fully connected layers [28], the average pooling [34] layer is used. Then, there are two sibling output layers: one that produces softmax probability estimates over K classes and another layer that produces bounding boxes for the predicted class. This is realized by a 3×3 convolutional layer followed by sibling 1×1 convolutions for dimensionality reduction. Default boxes are similar to anchor boxes used in Faster R-CNN [26], however we apply them to several feature maps of different resolutions. The final class prediction is given by the softmax output.

TABLE I
THE ARCHITECTURE OF TFFNET.

Backbone	MobileNet/VGG-16/Inception/ResNeXt, C				
EFP	global average pooling, C				
	d1_conv3_1	d2_conv3_2	d3_conv3_4	d4_conv3_8	d5_conv3_16
	3×3 d_1,1, d_1	3×3 d_2, 2, d_2	3×3 d_3, 4, d_3	3×3 d_4, 8, d_4	3×3 d_5, 16, d_5
	forward feature fusion, $1 \times 1, C$				
predictor	average pooling, C				
	$\begin{Bmatrix} 1 \times 1, 128 \\ 3 \times 3, 256 \end{Bmatrix}$	$\begin{Bmatrix} 1 \times 1, 128 \\ 3 \times 3, 256 \end{Bmatrix}$	$\begin{Bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \end{Bmatrix}$	$\begin{Bmatrix} 1 \times 1, 256 \\ 3 \times 3, 512 \end{Bmatrix}$	$\begin{Bmatrix} 1 \times 1, 1024 \\ - \end{Bmatrix}$
	softmax				

A. EFP module

EFP module is based on a factorized form of convolution that splits a standard convolution into a point-wise convolution and a feature pyramid of dilated convolutions. Unlike a serial-shaped FP [26]–[28] based on padding and stride convolutions, the EFP module is a parallel-shaped FP based on dilated convolutions.

Let a backbone network (such as MobileNet [35], VGG-16 [18], Inception [36], or ResNeXt [34]) produce an C-dimensional feature pyramid pool, $\boldsymbol{f} \in \mathbb{R}^{N\times M\times C}$, where N and M stand for the width and height of the feature map and C is the number of feature channels. Features are extracted from the final convolutional layer of a specific stage [25], [26], such as the *block5_conv3* layer of VGG-16 [18] and the *"conv2d_11"* layer of MobileNet [35]. Before feeding $\boldsymbol{f}$ into the EFP, we perform global average pooling [35], [37] on each channel of $\boldsymbol{f}$.

For efficient parallel computing, we first apply the point-wise convolution, and then the C-dimensional space is split into k branches with $d = \lfloor \frac{C}{k} \rfloor$ feature channels in each branch. The kernel of the point-wise convolution is of size $1 \times 1 \times C$. We create d kernels of size $1 \times 1 \times C$ to obtain a branch feature map $\boldsymbol{f}^p$ of size $N \times M \times d$, where $p = 1, \ldots, k$ is a level of the pyramid. In the splitting step, each branch uses the point-wise convolution to transform a high-dimensional feature map $\boldsymbol{f}$ into several low-dimensional feature maps, $F = \{\boldsymbol{f}^1, \boldsymbol{f}^2, \ldots, \boldsymbol{f}^k\}$. This convolution layer has $d \cdot 1 \cdot 1 \cdot C = Cd$ parameters.

EFP focuses on enhancing features of each pyramid level. Dilated convolutions [38] can exponentially expand the receptive field without losing resolution or coverage, and they are formulated as

$$(F *_p \kappa)(\boldsymbol{p}) = \sum_{\boldsymbol{f}+p\boldsymbol{t}=\boldsymbol{p}} F(\boldsymbol{f})\kappa(\boldsymbol{t}), \tag{17}$$

where $*_p$ denotes a p-dilated convolution and κ is a discrete $n\times n$ kernel. The effective receptive field of a p-dilated convolutional kernel is $[(n-1)2^p + 1]^2$. Each branch simultaneously resamples the feature maps of pyramid levels using d $n \times n \times d$ dilated convolutional kernels with different dilation rates 2^p. The dilated convolution layer has $k \cdot d \cdot n \cdot n \cdot d = kn^2d^2$ parameters. The output $P = \{\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots, \boldsymbol{p}^k\}$ of these k pyramid levels provides the contextual information at different channel scales

$(n-1)2^p d$. Since d is the rounded down integer, if the outputs of different dilated convolutions are directly concatenated, the unwanted gridding artefact will appear in the final output feature maps.

To remove the gridding artefacts, each branch utilizes a form of aggregating transformation [34] to merge the forward feature maps F into the enhanced feature maps P, referred to as a forward feature fusion. The outputs $Q = \{\boldsymbol{q}^1, \boldsymbol{q}^2, \ldots, \boldsymbol{q}^k\}$ from the feature fusion are hierarchically concatenated [36] to generate an EFP's output $\boldsymbol{q}$ with K feature channels, which have $k \cdot d \cdot 1 \cdot 1 \cdot K = kdK$ parameters.

To summarize, the EFP effectively aggregates the multi-scale context information via widening the network instead of increasing its depth, and it has $Cd + kn^2d^2 + kdK$ parameters. It is compared with similar convolution modules in TABLE II. The EFP and ResNeXt have higher effective receptive fields than MobileNet and Inception. Unlike the ResNeXt, the EFP is computationally more efficient and has lower memory requirements.

TABLE II
COMPARISON OF DIFFERENT CONVOLUTIONAL MODULES.

Module	of Number of parameters	Memory size	Receptive Field
MobileNet	$C(n^2+K)$	$(C+K)NM$	$[n]^2$
Inception	$k(Cd+n^2d^2)$	$2kNMd$	$[n]^2$
ResNeXt	$k(Cd+n^2d^2+dK)$	$kNM(2d+K)$	$[(n-1)2^{k-1}+1]^2$
EFP	$Cd+kn^2d^2+kdK$	$NMd(k+1)$	$[(n-1)2^{k-1}+1]^2$

B. Forward Feature Fusion

To match sizes of the feature maps, we resample the feature maps F and P with a series of $1 \times 1 \times K$ convolutions. The forward feature fusion is computed as:

$$\boldsymbol{q}^i = \sum_{i=0}^{K-1} [\boldsymbol{w}_i \boldsymbol{f}^i + (\mathbf{1} - \boldsymbol{w}_i)\boldsymbol{p}^i], \tag{18}$$

where $\boldsymbol{w}_i$ is a weight for the i-th branch, defined as

$$\boldsymbol{w}_i = \frac{\exp(\boldsymbol{z}^i)}{\sum_{j=0}^{K-1} \exp(\boldsymbol{z}^j)}, \tag{19}$$

where $\boldsymbol{z}^i = (\boldsymbol{f}^i)^T \boldsymbol{p}^i$ are activated elements and $\boldsymbol{f}^i$ is the low-dimensional feature maps.

IV. APPLICATION TO UNDERWATER SIGNALS

The performance of the TFFNet with EFP and sparse ACT is evaluated on two UWA datasets and compared with state-of-the-art networks. These datasets are UWA communication signal dataset and Whale FM sound dataset.

A. Two datasets and preprocessing

Underwater acoustic communication signal dataset: The UWA communication data is collected at different locations in the Wuyuan Bay, Xiamen, China, from 2016 to 2018. Two *universal deck devices* (http://www.soundlong.cn/Product/348012305.html) are used to send and receive underwater communication signals at frequencies from 20 to 40 kHz. The distance between communication nodes is in the interval from 0.1 km to 10 km.

The signal dataset consists of 2500 training, 550 validation, and 350 test audio files in WAV format ranging between 2.6 s and 42.3 s in length. The signals were sampled at 60, 66, 96, 75 or 156 kHz. There are five modulation types, labelled as MCMFSK (Multi-carrier Multiple Frequency-shift Keying), OFDM (Orthogonal Frequency Division Multiplexing), CIOFDM (Carrier Interferometry, CI), LFM_OFDM (Linear Frequency Modulation, LFM) and DFT_OFDM (Discrete Fourier Transform, DFT), and transmitted messages in all modulation signals, labelled as Data. There are 680 signals with each type of modulation, of which 500, 110, and 70 are used for training, validation and testing, respectively. The class 'Data' represents the carried message in the modulation signal, and all modulation signals contain the type of 'Data' such that size of its training, validation and test sets are 2500, 550, 350, respectively.

Short term energy (STE), zero crossing rate (ZCR) and fast permutation entropy (FPE) [39] are first computed to select signal segments for training. These methods are applied for the semi-automated annotation of the unlabelled data in the UWA communication signal dataset. Fig. 3 shows an example to elaborate the preprocessing of signals. A communication signal generally contains a guide segment, data segment and invalid data segment (marine environmental noise), see Fig. 3(e). The STE associated with the invalid segment is small when compared to the guide segment and data segment, see Fig. 3(b). Thus, in the preprocessing stage, the invalid segment can be removed by referring to the value of STE. When the ZCR is high, it may be inferred that the signal segment contains high-frequency components. This phenomenon is illustrated in Fig. 3(c). The

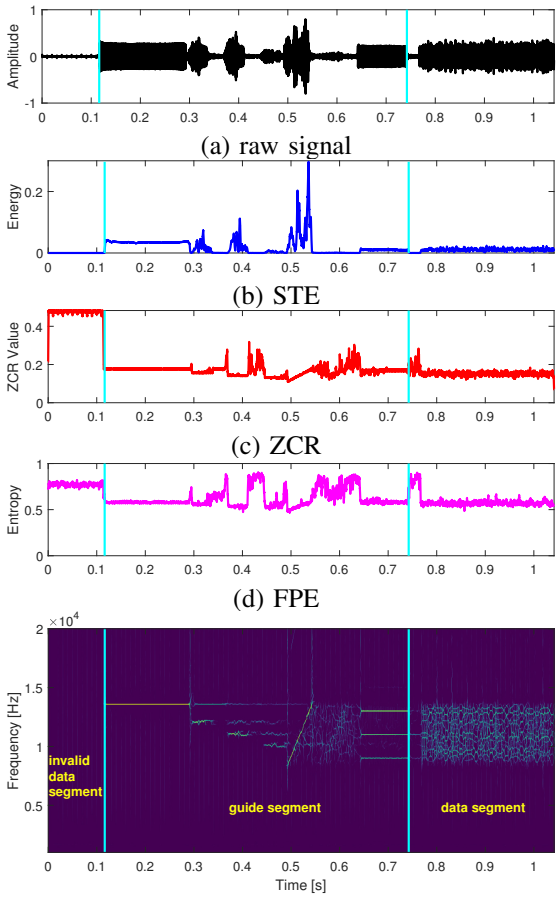


Fig. 3. An UWA communication signal represented by STE, ZCR, FPE and sparse ACT. The drawn lines represent start and end points of a guide segment.

FPE is a complexity measure for chaotic time series [39]. FPE sometimes indicates transitions that are not properly detected by ZCR, see Fig. 3(d). Segments of interest are selected to create signal labels for data segments using ZCR and FPE.

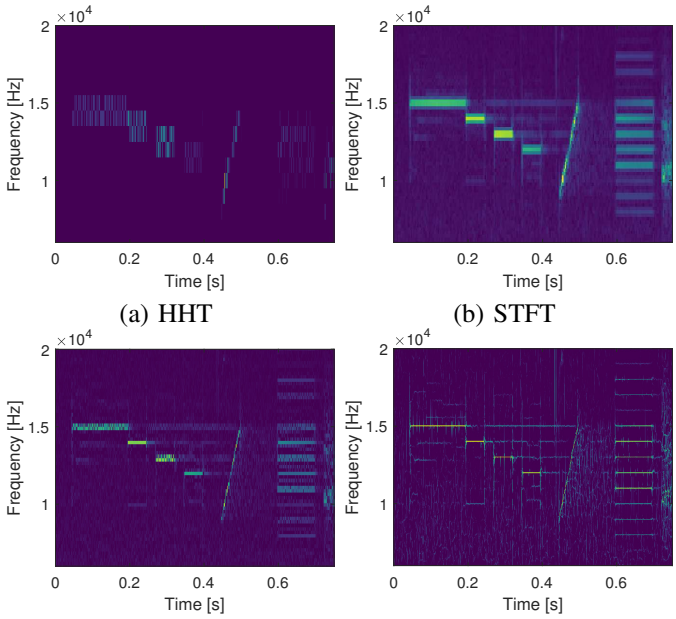
Whale FM sound dataset: The Whale FM dataset is available from the website: https://github.com/zooniverse/WhaleFM [13]. A 'whale_fm_anon_04-03-2015_assets.csv' is used to extract the data. The following are some of the important columns in the file: 'name', 'location', 'whale_id', and 'whale_type'. In this work, we select 15500 audio files of 1 s to 8 s long. They are produced by 15 killer whales and 11 pilot whales at locations off the coast of Norway, Iceland, and Bahamas. All audio files are divided into 2 classes (pilot and killer whales) by 'whale_type' and 26 classes (5-14, 22-26 classes of killer whales and 1-4, 5-21 classes of pilot whales) by 'whale_id'. Sampling rates of resampled audio files are 22.05 kHz, 32 kHz and 48 kHz. The dataset of each class is divided into training (60%), validation (20%) and test sets (20%) such that each set has similar class distribution.

B. Experiments

All of the networks are trained by using four-core Intel(R) Core(TM) i7-8700 CPU and NVIDIA GeForce GTX 1050 Ti GPU, and Tensorflow (https://www.tensorflow.org/) in Python with CUDA 9.2 and cuDNN back-ends (https://developer.nvidia.com/). An inverse class probability weighting is introduced in the cross-entropy loss function to solve the class imbalance [28]. To update parameters of the networks during training, the ADAM optimizer [40] was utilized with an initial learning rate of 5×10^{-4} and with a weight decay of 5×10^{-3}. The fork between training and validation performance commonly adopts 32 epochs [26].

In the work, four parameters σ_t, σ_f, θ and λ of the sparse ACT are learnt from UWA signals. For the sparse ACT, the angle list ranges from $\frac{\pi}{4}$ to π with an interval of $\frac{\pi}{8}$. Parameters σ_t and σ_f each is a list of values [0.2, 0.5, 0.8, 1.1, 1.4, 1.7, 2.0, 2.3]. For high computational efficiency, σ_t and σ_f should be large values, e.g., a list [0.5, 0.5+Δ, $\cdots$, 5] of σ_t, σ_f values with a large $\Delta \in [1, 2.2]$. For the high energy concentration, σ_t and σ_f should be small values, e.g., a list $[0.1, 0.1 + \Delta, \cdots, 3]$ of σ_t, σ_f values with $\Delta \in [0.2, 1]$. The work [12] has demonstrated that when the parameter λ is set to 2, the TFR with the highest energy concentration is achieved. The parameter $\lambda = 2$ is selected.

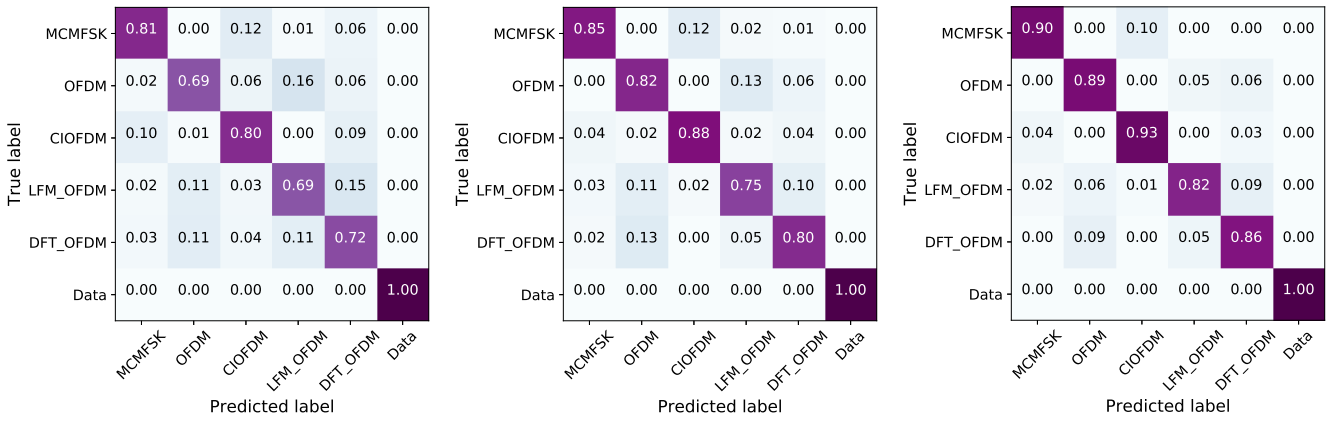
Experiment 1: In order to demonstrate that the sparse ACT is an efficient TF representation for the observed UWA signals, several TF representations such as the short-time Fourier transform (STFT), Hilbert-Huang transform (HHT) [41] and Fourier synchroSqueezing transform (FSST) [9] are also considered for comparison.



(a) HHT (b) STFT

(c) FSST (d) Sparse ACT

Fig. 4. TFRs of a MCMFSK signal obtained by using HHT, STFT, FSST and sparse ACT.



(a) HHT (b) STFT (c) Sparse ACT

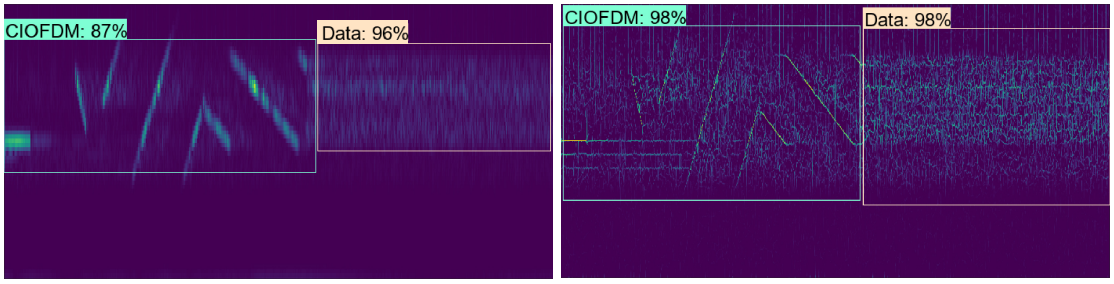
Fig. 5. Confusion matrix for TFFNet with TF feature obtained by HHT, STFT and sparse ACT on UWA communication signals. 'Data' class denotes the transmitted message of all modulated signals. It depicts per-class accuracy obtained via the use of the TFFNet.

Consider an MCMFSK signal, the sampling frequency is 200 kHz. The observed signal has a length of 150000 samples within a time interval of 0.75 seconds. The STFT and FSST are calculated with a Kaiser window of length 900. For the STFT, the number of overlapped samples is 220. Fig. 4 illustrates that the sparse ACT reveals distinctive features of an MCMFSK signal that cannot be seen in HHT, STFT and FSST.

Experiment 2: To quantify the benefits of using ACT in the TFFNet architecture, we perform the modulation type classification.

Fig. 5 compares the classification performance of HHT, STFT and sparse ACT with the TFFNet network in application to the UWA communication signals test dataset. Three TFFNet classifiers estimate classification confidence scores [28], [42] of the predicted class on the test dataset. For five modulation classes, the confusion matrix for the sparse ACT shows higher classification confidence scores than HHT and STFT.

We provide an example of test set to visually verify two trained TFFNet classifiers with STFT and sparse ACT. Fig. 6 shows that two trained TFFNet classifiers detect guide and data segments of the CIOFDM signal. For the TFFNet classifier based on sparse ACT, the classification scores of guide and data segment are 98% and 98%, which are higher than the scores 87%



(a) STFT (b) Sparse ACT

Fig. 6. The TFR ($t \in [0, 0.9993]$ s, $f \in [0, 10]$ kHz)) of a CIOFDM signal is an example on test dataset and is used to test two trained TFFNet classifiers. A CIOFDM signal contains two classes, CIOFDM class (aquamarine) and 'Data' class (bisque). 'CIOFDM:98%' expresses the detected class and its classification score.

and 96% when using the TFFNet classifier based on STFT. Since higher resolution TFRs obtained by the sparse ACT provide better quantifications of UWA signals, the TFFNet classifier with sparse ACT is a superior alternative to networks using STFT spectrogram.

TABLE III
AVERAGE PRECISION FOR EACH CLASS AND CLASSIFIER' MAP FOR THE UWA COMMUNICATION SIGNALS DATASET.

Model	backbone	MCMFSK	OFDM	CIOFDM	LFM_OFDM	DFT_OFDM	mAP
SWCNN	MobileNet	0.726	0.603	0.699	0.708	0.516	0.651
	VGG-16	0.733	0.659	0.702	0.713	0.521	0.666
	Inception-V3	0.812	0.691	0.806	0.756	0.598	0.733
	ResNeXt-50	0.834	0.703	0.812	0.793	0.601	0.749
	ResNeXt-101	0.901	0.756	0.888	0.8	0.634	0.796
RCNN	MobileNet	0.735	0.669	0.732	0.791	0.635	0.713
	VGG-16	0.798	0.788	0.849	0.853	0.711	0.8
F-RCNN	MobileNet	0.851	0.783	0.831	0.813	0.725	0.801
	VGG-16	0.914	0.874	0.902	0.899	0.796	0.877
	ResNeXt-50	0.873	0.832	0.851	0.896	0.753	0.841
R-FCN	ResNeXt-50	0.905	0.812	0.866	0.825	0.736	0.829
	ResNeXt-101	**0.927**	0.861	0.9	0.913	0.809	0.882
SSD	VGG-16	0.903	0.877	0.89	0.902	0.846	0.884
	ResNeXt-101	0.915	0.9	0.883	**0.915**	0.857	0.894
TFFNet	VGG-16	0.869	0.854	0.903	0.863	0.869	0.872
	Inception-V3	0.926	**0.903**	**0.958**	0.913	**0.906**	**0.921**
	ResNeXt-50	0.893	0.88	0.932	0.905	0.89	0.9

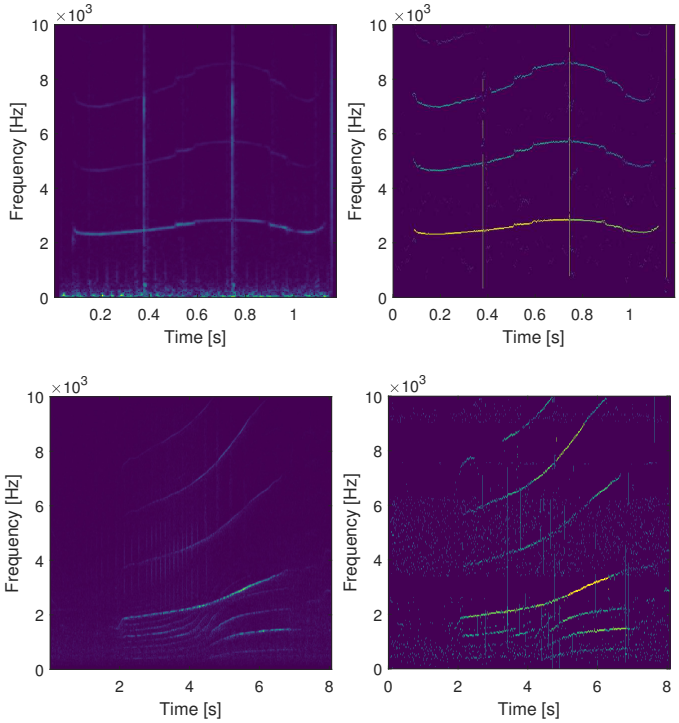
Experiment 3: For multiple object classification, the traditional SWCNN, region-based networks (RCNN [19], F-RCNN [27], R-FCN [25]) and region-free SSD [28] achieve good performance on many datasets. Therefore, we compare the performance of TFFNet with these networks on the UWA communication signal dataset. The efficient convolutional modules: MobileNet [35], VGG [18], Inception V3 [36], and ResNeXt [34] are used as a backbone of these networks, where the input size of TF images is 300×300. MobileNet [35], VGG-16 [18] and Inception V3 [36] extract features from the named "conv2d_11", "conv5_3" and "Mixed_6e" layers, respectively. ResNeXt-50 and ResNeXt-101 [34] extract features from the last layer of the "conv4" block. For a fair comparison, we fine tune *convolutional* layers and use the same parameter settings of these modules.

TABLE III shows the classification performance of TFFNet and other networks with various backbone modules. The mAP is the mean average precision of the classes [27]. SWCNN and RCNN achieve lower mAP than the F-RCNN or R-FCN network. For the traditional SWCNN, the disadvantage of using a fixed sized window is that thin components may not suit within the window. We observe that the total mAP of F-RCNN achieves its highest peak 0.877 when using VGG16. R-FCN with ResNeXt-101 yields a slightly better performance over F-RCNN. SSD networks show superior performance than most of the region-based methods. TFFNet with Inception V3 achieves the highest mAP on this task. The experimental results demonstrate that the classification performance can be enhanced by using the proposed TFFNet, when classifying UWA communication signals.

Experiment 4: The fourth experiment focuses on two tasks, classification of pilot and killer whales and classification of the individual whales.

Fig. 7 is an example of Whale FM spectrograms for impulsive and tonal events. The sound of the pilot whale at the top of Fig. 7 has impulsive and tonal components, which are overlapped in the TF image. The sound of the killer whale at the bottom of Fig. 7 has dominant tonal components. Fig. 7(a) shows the STFT of the original signal, calculated with the Hanning window of length 256. The TF features of STFT appear to be blurry and even some features are missing in the TFR. As can

be seen in Fig. 7(b), TFR using the sparse ACT has higher energy concentration than TFR using STFT. It can be concluded that, the sparse ACT is more suitable for analysis of whale signals than STFT.



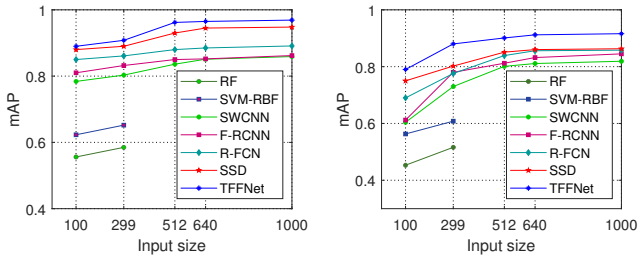
(a) STFT (b) Sparse ACT

Fig. 7. Example TFRs of sounds of a Norwegian pilot whale (top) and a Norwegian killer whale (bottom).

TABLE IV
CLASSIFICATION MAP OF 26-CLASS WHALES.

Model	TFR	Input size	mAP(%)	Model	TFR	Input size	mAP(%)
TFFNet - EFP	ACT	299×299	80.5%	TFFNet - EFP	STFT	299×299	68.9%
TFFNet - EFP	ACT	512×512	82.7%	TFFNet - EFP	STFT	512×512	72.6%
TFFNet + EFP	ACT	299×299	83.5%	TFFNet + EFP	STFT	299×299	69.3%
TFFNet + EFP	ACT	512×512	90.6%	TFFNet + EFP	STFT	512×512	75.8%

Detecting thin tonal and pulsed structures is one of the most challenging problems for both the region-based CNN (R-FCN) and region-free methods (SSD) [19], [28]. To assess the benefits of using the EFP module, we perform an experiment with two settings: TFFNet using Inception V3 backbone with and without EFP for the classification of 26-class whales. TABLE IV demonstrates that the designed TFFNet with EFP, providing the context information at different scales, yields a significant improvement in the classification on the Whale FM dataset. Furthermore, the size and resolution of input images are two key factors influencing the classification accuracy. For the TFFNet classifier with ACT, enlarging the size of input TF images to 512×512 improves mAP to 90.6%, compared to 83.5% with a size 299×299. Since TFR images obtained by the STFT have lower resolution than the ACT, TFFNet with STFT results in a lower mAP than the TFFNet with ACT.



(a) 2-class (b) 26-class

Fig. 8. Performance comparison for different classification tasks.

A goal is to illustrate the phenomenon of improving the accuracy as the number of classes and input image size increase. Fig. 8 demonstrates the classification performance of ResNeXt-50 backbone on 2-class (pilot and killer whales) and 26-class (11 pilot and 15 killer whales) classification tasks, respectively. The results show that TFFNet with EFP achieves the highest classification on both tasks and the increase of the input image size leads to an improved classification performance.

The next example, for a pilot whale, is illustrated in Fig. 9, showing the TF image obtained using the sparse ACT. It shows that the trained TFFNet classifier detects the acoustic event, PW_15, in the sound. The highest classification score of PW_15 event is 95%. Therefore, the trained TFFNet classifier has good classification performance to discriminate the acoustic events of the whale sound.

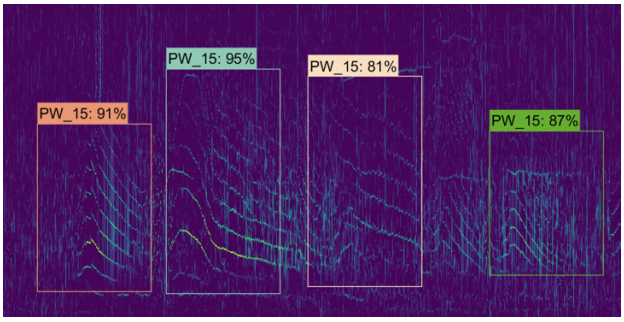
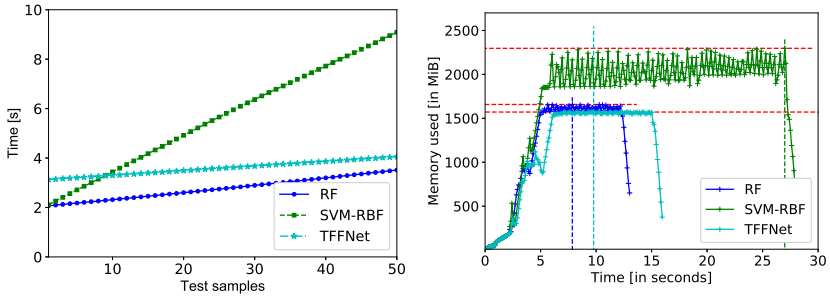


Fig. 9. Example of qualitative classification of a pilot whale with TFFNet. Each color corresponds to an event category in the TF image ($t \in [0, 8.945]$ s, $f \in [0, 12]$ kHz). PW_{·} expresses a label where PW is an abbreviation for a particular pilot whale.



(a) (b)

Fig. 10. Testing time and memory of three classifiers trained by RF, SVM-RBF and TFFNet. (a) Time consumption of 50 signal samples classification. (b) Memory profiler report from start to finish. The vertical lines denote the time point of the maximum memory consumption.

Experiment 5: The classification performance of the TFFNet is compared with that of two machine learning methods, random forest (RF) and support vector machine with radial basis function kernel (SVM-RBF). RF and SVM-RBF are implemented in Python with the sklearn (https://scikit-learn.org/) package. In order to achieve high accuracy by SVM, the RBF kernel is approximated in a high dimensional space by embeddings, but grid search to determine parameters of the RBF will increase the computation time of the SVM-RBF. Due to a problem of hardware capacity to process large images in the SVM-RBF, an TF image of a size 299×299 is divided into 9 patches as a group of inputs.

For the classification of Whale FM datasets, the results shown in Fig. 8 demonstrate that TFFNet achieves higher classification accuracy than RF and SVM-RBF for both 2-class and 26-class tasks. The RBF classification is less accurate compared to SVM-RBF. In addition, 50 test samples are classified by using three trained classifiers, namely RF, SVM-RBF and TFFNet with an input size 100×100. Fig. 10 provides their test time and memory. Loading a trained TFFNet model into memory takes more time than loading RF and SVM-RBF. The time of the classification processing is lower than that of RF and SVM-RBF. In Fig. 10(b), the memory profiler report visualizes counters that represent the total allocated memory of three trained classifiers during the running time. It has been observed that the memory consumption of TFFNet is less than that of RF and SVM-RBF.

V. Conclusions

In this paper, we have proposed a sparse anisotropic chirplet transform for producing an TFR of UWA signals. The benefits of using the sparse ACT in analyzing underwater signals were verified in a number of experiments. The sparse ACT generates a high-resolution TFR so that it is capable of revealing fine TF features that would otherwise be hidden if analyzed using the traditional STFT. It is a useful tool in analyzing acoustic events, for discrimination and classification of differences between signals.

We have also proposed TFFNet, a fine-tuned model generalizing convolutional neural networks to TF domains, permitting to accomplish deep learning on underwater sound data. The proposed efficient feature pyramid with forward feature fusion preserves useful information on signals and improves the feature discrimination.

When compared with classifiers based on other recently proposed advanced backbone networks, TFFNet achieves superior classification performance on UWA communication signal dataset and Whale FM sound dataset, especially with thin or linear structures in TF images. By employing the efficient feature pyramid, TFFNet exhibits a great performance on both 2-class and 26-class tasks. The TFFNet compared with two machine learning methods, random forest (RF) and support vector machine with radial basis function kernel (SVM-RBF) achieves higher classification accuracy, and requires lower memory and classification time.

ACKNOWLEDGMENT

The work of Yongchun Miao and Haixin Sun was partly supported by the National Natural Science Foundation of China (61971362) and the National Key R&D Program of China (2018YFC0809200). The work of Y. Zakharov was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through the USMART (EP/P017975/1) and Full-Duplex (EP/R003297/1) projects.

REFERENCES

[1] B. G. Ferguson, "Time-frequency signal analysis of hydrophone data," *IEEE Journal of Oceanic Engineering*, vol. 21, no. 4, pp. 537–544, October 1996.
[2] R. Vergin, D. O'Shaughnessy, and A. Farhat, "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 525–532, September 1999.
[3] R. Diamant, "Closed form analysis of the normalized matched filter with a test case for detection of underwater acoustic signals," *IEEE Access*, vol. 4, pp. 8225–8235, November 2016.
[4] W. Li, S. Zhou, P. Willett, and Q. Zhang, "Preamble detection for underwater acoustic communications based on sparse channel identification," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 256–268, January 2019.
[5] B. Li, M. Sun, X. Li, A. Nallanathan, and C. Zhao, "Energy detection based spectrum sensing for cognitive radios over time-frequency doubly selective fading channels," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 402–417, January 2015.
[6] S. Mann and S. Haykin, "The chirplet transform: physical considerations," *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2745–2761, November 1995.
[7] Y. Miao, H. Sun, and J. Qi, "Synchro-compensating chirplet transform," *IEEE Signal Processing Letters*, vol. 25, no. 9, pp. 1413–1417, September 2018.
[8] G. K. Nilsen, "Recursive time-frequency reassignment," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3283–3287, August 2009.
[9] F. Auger, P. Flandrin, Y. Lin, S. McLaughlin, S. Meignen, T. Oberlin, and H. Wu, "Time-frequency reassignment and synchrosqueezing: An overview," *IEEE Signal Processing Magazine*, vol. 30, no. 6, pp. 32–41, November 2013.
[10] F. Boßmann and J. Ma, "Asymmetric chirplet transform for sparse representation of seismic data," *Geophysics*, vol. 80, no. 6, pp. WD89–WD100, November 2015.
[11] Y. Miao, H. Sun, and J. Qi, "Intrinsic mode chirp multicomponent decomposition with kernel sparse learning for overlapped nonstationary signals involving big data," *Complexity*, vol. 2018, no. 8426790, pp. 1–15, July 2018.
[12] Y. Miao, H. Sun, and J. Wang, "Anisotropic instantaneous frequency estimator," in *IEEE International Conference on Signal Processing, Communications and Computing*, Dalian, China, September 2019, pp. 1–5.
[13] L. Shamir, C. Yerby, R. Simpson, A. Von Benda-Beckmann, P. Tyack, F. Samarra, P. Miller, and J. Wallin, "Classification of large acoustic datasets using machine learning and crowdsourcing: application to whale calls," *Journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 953–962, February 2014.
[14] V. Mitra, C.-J. Wang, and S. Banerjee, "Lidar detection of underwater objects using a neuro-svm-based architecture," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 717–731, May 2006.
[15] M. Elforjani and S. Shanbr, "Prognosis of bearing acoustic emission signals using supervised machine learning," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5864–5871, July 2018.
[16] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, October 2001.
[17] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *MIT press*, pp. 109–129, 2016.
[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[19] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, September 2014, pp. 580–587.
[20] L. Zhang, D. Wang, C. Bao, Y. Wang, and K. Xu, "Large-scale whale-call classification by transfer learning on multi-scale waveforms and time-frequency features," *Applied Sciences*, vol. 9, no. 5, p. 1020, March 2019.
[21] M. Zhong, M. Castellote, R. Dodhia, J. Lavista Ferres, M. Keogh, and A. Brewer, "Beluga whale acoustic signal classification using deep learning neural network models," *The Journal of the Acoustical Society of America*, vol. 147, no. 3, pp. 1834–1841, March 2020.
[22] O. S. Kirsebom, F. Frazao, Y. Simard, N. Roy, S. Matwin, and S. Giard, "Performance of a deep neural network at detecting north atlantic right whale upcalls," *The Journal of the Acoustical Society of America*, vol. 147, no. 4, pp. 2636–2646, April 2020.
[23] Y. Shiu, K. Palmer, M. A. Roch, E. Fleishman, X. Liu, E.-M. Nosal, T. Helble, D. Cholewiak, D. Gillespie, and H. Klinck, "Deep neural networks for automated detection of marine mammal species," *Scientific Reports*, vol. 10, no. 1, pp. 1–12, January 2020.
[24] M. Kubanek, J. Bobulski, and J. Kulawik, "A method of speech coding for speech recognition using a convolutional neural network," *Symmetry*, vol. 11, no. 9, p. 1185, September 2019.
[25] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017.
[26] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, April 2018.
[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.
[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, pp. 21–37, December 2016.
[29] Y. Miao, H. Sun, and J. Qi, "Synchro-compensating chirplet transform," *IEEE Signal Processing Letters*, vol. 25, no. 9, pp. 1413–1417, September 2018.

[30] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1068–1089, May 1995.
[31] K. Kodera, R. Gendrin, and C. Villedary, "Analysis of time-varying signals with small BT values," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 64–76, February 1978.
[32] J. . Geusebroek, A. W. M. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–943, August 2003.
[33] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113–125, February 1998.
[34] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, November 2017, pp. 1492–1500.
[35] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, April 2017.
[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 2818–2826.
[37] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.
[38] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.
[39] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical review letters*, vol. 88, no. 17, p. 174102, April 2002.
[40] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research*, vol. 12, pp. 2121–2159, July 2011.
[41] E. Elbouchikhi, V. Choqueuse, Y. Amirat, M. E. H. Benbouzid, and S. Turri, "An efficient hilbert-huang transform-based bearing faults detection in induction machines," *IEEE Transactions on Energy Conversion*, vol. 32, no. 2, pp. 401–413, June 2017.
[42] Mingkun Li and I. K. Sethi, "Confidence-based active learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1251–1261, August 2006.